DOCUMENT RESUME

ED 152 229 IR 005 509

AUTHOR Anderson, John R.; Kline, Paul J.
TITLE Design of a Production System for Cognitive Modeling #1. Technical Report 77-2.
INSTITUTION Yale Univ., New Haven, Conn. Dept. of Psychology.
SPONS AGENCY Office of Naval Research, Arlington, Va. Personnel and Training Research Programs Office.
PUB DATE 77
CONTRACT N0014-77-C-0242
NOTE 30p.; Best copy available

EDRS PRICE MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS *Artificial Intelligence; *Cognitive Processes; Computational Linguistics; Computer Programs; Decision Making; *Information Processing; *Models; *Programing Languages; Thought Processes
IDENTIFIERS *ACT

ABSTRACT

This report describes several of the design decisions underlying ACT, a production system model of human cognition. ACT can be considered a high level computer programming language as well as a theory of the cognitive mechanisms underlying human information processing. ACT design decisions were based on both psychological and artificial intelligence considerations. This is discussed in terms of: (1) the data base as a propositional network, (2) spreading activation, (3) the data driven character of production systems, (4) unity of control and data store, (5) modularity, and (6) parallelism. (Author/STS)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

ED152229

I R

# Design of a Production System for Cognitive Modeling #1

U S DEPARTMENT OF HEALTH
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

John R. Anderson

Paul J. Kline

Department of Psychology
Yale University
New Haven, Connecticut 06520

Approved for public release; distribution unlimited.
Reproduction in whole or in part is permitted for
any purpose of the United States Government.

This research was sponsored by the Personnel and Training Research Programs, Psychological Services Division, Office of Naval Research, under Contract No.: N00014-77-C-0242, Contract Authority Identification Number, NR No. 154-399.

BEST COPY AVAILABLE

IR005509

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>Technical report 77-2 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>Design of a Production System for Cognitive Modelling #1 | | 5. TYPE OF REPORT & PERIOD COVERED<br>technical report<br>(June 1, 1977 - September 1, 1977)<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>John R. Anderson and Paul J. Kline | | 8. CONTRACT OR GRANT NUMBER(s)<br>N00014-77-C-0242 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Department of Psychology<br>Yale University<br>Box 11a Yale Station<br>New Haven, CT 06520 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>66153N<br>RR042-04-01 NR 154-399 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Personnel and Training Research Programs,<br>Office of Naval Research (Code 458)<br>Arlington, VA 22217 | | 12. REPORT DATE<br>September 1, 1977<br>13. NUMBER OF PAGES<br>36 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>Unclassified<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report)<br>Approved for public release; distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number)<br>Artificial intelligence<br>Associative networks<br>Cognitive Psychology<br>Computer simulation<br>Learning<br>Memory<br>Production system | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) | | |

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

It is conjectured that a good cognitive psychology theory will lead to a good artificial intelligence (AI) program. If this is true there should be a convergence of psychological and AI considerations in theory construction. This convergence is illustrated in terms of ACT, a computer simulation model of cognitive processes. Separate AI and psychological considerations are used to motivate the decision to design ACT as a production system operating on an associative network data base. Similar motivation is provided for other features of ACT implemented within this framework. These features include the use of a propositional structure for the associative network, a spreading activation process operating on the network, the simulated ability to execute several procedures in parallel, and the use of strength measures to select among competing productions and competing paths in the network.

Abstract

It is conjectured that a good cognitive psychology theory will lead to a good artificial intelligence (AI) program. If this is true there should be a convergence of psychological and AI considerations in theory construction. This convergence is illustrated in terms of ACT, a computer simulation model of cognitive processes. Separate AI and psychological considerations are used to motivate the decision to design ACT as a production system operating on an associative network data base. Similar motivation is provided for other features of ACT implemented within this framework. These features include the use of a propositional structure for the associative network, a spreading activation process operating on the network, the simulated ability to execute several procedures in parallel, and the use of strength measures to select among competing productions and competing paths in the network.

We have been working on a production system model of human cognition called ACT. An earlier version of the ACT system, called ACTE, is described in Anderson [[3]], Anderson, Kline, and Lewis [[5]], and Kline and Anderson [[21]]. That system has been used to develop mini-models for retrieval from memory, inference making, language comprehension, question-answering, and problem solving. We are currently working on a new version of ACT called ACTF. This paper discusses a number of the design decisions underlying the ACT system. We will discuss how these design decisions are motivated by both psychological and artificial intelligence (AI) considerations.

[What is ACT?]

ACT is at the same time a high-level programming language and a theory of the cognitive mechanisms underlying human information processing. A high-level programming language is a formalism that facilitates programming certain kinds of algorithms. However, it may also be difficult to program algorithms other than the intended kinds, so that high-level programming languages are often "special-purpose". ACT is a special-purpose programming language in this sense. But the fact that certain processes or algorithms can be coded in ACT more efficiently and easily than others is the means by which ACT provides a theory of the cognitive mechanisms that underlie human information processing. Humans are also more successful at certain cognitive processes than others. The hope is that ACT limitations correspond to human limitations. If this is so, psychological theories of specific

cognitive behaviors automatically acquire a certain face validity when embodied as programs in the ACT programming language.

Given the adaptiveness and flexibility of human cognition no static ACT program can serve as an adequate psychological model. ACT must have the capability for evolving new programs and our current work is focused on developing a learning program. However, space limitations do not permit us to discuss this learning work, we will confine the discussion to the principles by which ACT programs are interpreted and executed.

[A Relationship Between Psychology and AI]

We conjecture that there is a strong relationship between cognitive psychology and those AI efforts concerned with developing general and adaptive systems. (This is what we will mean by AI for the remainder of this paper.) This conjecture can be stated as follows:

Good cognitive psychology is good artificial intelligence. (1)

The majority of this paper will be devoted to illustrating how belief in this principle has influenced the design of the ACT system. However, first we would like to discuss the implications of this claim.

What exactly does it mean? By "good psychology" we mean a theory that meets four scientific criteria: parsimony, effectiveness, broad generality, and empirical accuracy (see [[3]] for a discussion of these). Of principal importance to our points are the criteria of effectiveness and empirical accuracy. Effectiveness refers to the constraint that the theory be specified well enough that predictions

can be rigorously derived from it. The important consequence of effectiveness is that a computer simulation of the theory can be produced . By empirical accuracy we mean that it successfully predicts human behavior in various cognitive tasks. Thus by conjecture (1) we mean that the computer simulation of a psychological theory that successfully accounts for empirical data will be translatable into a good AI program. By a good AI program we mean one that produces intelligent behavior under reasonable constraints of computational efficiency. Since a good psychology program will mimic human behavior, a reasonable standard of intelligence is guaranteed. The non-obvious aspect of our conjecture concerns the matter of computational efficiency.

If conjecture (1) is true, one way to attempt AI is to develop a program that models psychological data. Note, however, we are not claiming that this is the only way to develop a good AI theory. Moreover, we are explicitly not endorsing the claim that an AI program must meet standards of good psychology. That is, we are not claiming:

Bad cognitive psychology is bad artificial intelligence, (2)

nor its contrapositive equivalent:

Good artificial intelligence is good cognitive psychology. (3)

Claim (3) has been put forward (but not endorsed) by Newell [[25]] as a possible relationship between AI and psychology. It has been endorsed informally by a number of workers in AI. (We wonder if its endorsers were aware of its equivalent contrapositive (2) which would

make all AI endeavors subject to psychological test.) The argument for (3) is that the feats of human intelligence are so difficult that there is essentially only one way of accomplishing them on any physical device including a human brain or a serial computer. While what is meant by "essentially only one way" is uncertain, we feel the claim is implausible. We find this claim implausible in light of the non-identifiability results in psychology [[3]], the behavioral equivalences among different machines in automata theory, the existence of very different programs to perform a range of functions from sorting to parsing, and the observation that nature has evolved different mechanisms to serve the same function.

So, we feel there are different routes to the goal of good AI, only some of which involve psychological theories. For example, there would be nothing wrong with an AI program that found it easier to analyze a sentence than a picture, but there would be something wrong with a psychological theory that did. However, it is a fact that there is no current AI program that begins to meet the criteria of broad generality and adaptivity. Thus, while there may be many ways to achieve a good AI program, it clearly is proving difficult to discover any of these ways. Thus, while it is not necessary for the AI worker to follow psychological research in designing his systems, it might prove to be a good heuristic in trying to search for an adequate program.

The implications of conjecture (1) about the relationship

between psychology and AI are not one-sided. This can be seen by considering the contrapositive of (1):

Bad artificial intelligence is bad cognitive psychology. (4)

This places a new constraint on a psychological theory. It must be translatable into a good AI program. In particular, the implementation of a theory must obey certain constraints on computational efficiency. For instance, a theory of language processing would be judged inadequate if its best simulation produced comprehension times that were exponential functions of sentence lengths or which required exponentially increasing computational space.

We think complexity functions are better measures of computational efficiency than are absolute amounts of computation space or time. Judgments about what is an acceptable absolute amount are determined by current machine capabilities. It would be silly to cripple our psychological theory because current machine capabilities do not match those of the human brain. It would be particularly silly given the reasonable expectation that current limitations will be exceeded by many orders of magnitude over the next decades.

However, evaluating the efficiency of an algorithm in terms of complexity functions is not without its problems either. Complexity functions make no allowances for the fact that often only a restricted range of complexities are encountered in a practical problem. AI programs employing algorithms with superior performance on this

restricted range would be preferred even if the asymptotic performance of these algorithms was very poor.

With suitable allowances for this problem with complexity functions, we might understand (1) and (4) as asserting that a good psychology program will employ algorithms which are characterized by reasonable complexity functions. But, there are still problems. Suppose our best simulation predicted that a certain process took an exponentially increasing amount of time as a function of problem complexity. That would not be unacceptable if it could be shown that the best known computer algorithm also displayed this complexity function and that human processing time increased exponentially with problem complexity. It is also the case that human behavior often displays a poorer complexity function than the best known algorithm. For instance, consider the time it takes humans to identify a concept in a concept-identification task [[17]]. Their identification times are often linear in the number of possible hypotheses rather than logarithmic. However, humans can be trained to implement a logarthmic algorithm. The upshot of these qualifications is that (1) and (4) amount to the requirement that the best algorithm implementable in a psychological theory like ACT display the best possible complexity function on a serial computer. We would conjecture that humans are capable of achieving this same complexity function after suitable training.

Now that we finally understand what (1) and (4) should be taken

to mean, we see that for (1) and (4) to be valid the human brain and the serial computer must be subject to the same complexity constraints. That is, if either computing device can employ algorithms whose complexity functions are unattainable by the algorithms available to the other, (1) and (4) will probably be false.

The fact that humans can hand simulate computer programs suggests that the brain is not at a disadvantage in this respect. On the other hand, we know of no physically realizable computational device (the brain included) that achieves better complexity functions than the serial computer #2.

Thus barring the discovery that the brain has fundamentally superior computational abilities, (1) and (4) would appear to characterize the relationships between good psychology and good AI. If this characterization is accurate, there is potential for fruitful interaction between the two disciplines. The remainder of this paper will discuss how AI considerations and psychological considerations converge in the case of the ACT program.

[Propositional Network]

The data base in ACT is a propositional network. A propositional network is an associative network structured so it can be divided into units having a propositional status. There has been some concern whether associative networks have adequate expressive power (cf., [[35]]). However, in conjunction with the production system it can be shown that ACT has expressive power at least equivalent to the

predicate calculus [[3]]. The condition of an ACT production specifies network configurations that must be present or absent for the production to apply. The execution of an ACT production can cause new propositions to be built in memory and old propositions to be modified.

The associative network structure of ACT is such that each concept indexes all the propositions it occurs in. The structure is double-linked so it is possible to go both from concept to proposition and from proposition to concept. This associative indexing feature of ACT nicely illustrates the convergence of psychological and AI considerations. There is a large amount of data indicating some sort of associative structure in memory. A simple phenomenon is that of word association: Take a word, generate a semantic associate to it, take that word and generate an associate to it, and so on. For instance, starting with dog: dog-bark-hear-ear-organ-body-weight-scale-step-foot-ball-etc. The obvious way to simulate such free association protocols (cf., [[20]] ) is by a search process over an associative network. The computational advantage of associative indexing is also well understood. It serves to make time to retrieve a fact approximately independent of the number of facts in the data base.

There are also converging arguments for imposing a propositional structure on the data base rather than some simpler associative structure. Most AI programs that have performed inferential reasoning have used proposition-like representations of information. The reason is that a propositional structure abstracts

out and makes salient aspects of the information relevant for deciding the validity (or plausibility) of inferences.

There are numerous sources of psychological data supporting the concept of propositional organization. For instance, there is considerable evidence (e.g., [[6, 4, 16]]) that propositions tend to be forgotten as units. That is, the conditional probabilities are high that if one element of a proposition is forgotten the remainder will also be forgotten. While subjects do occasionally recall only partial propositions (e.g. "The hippie touched somebody but I can't remember who.") such reports are statistically rare. There is also considerable evidence (e.g., [[28, 32]]) that subjects, in remembering sentential information, often do not remember the exact wording of what was said but only the propositional content.

[Spreading Activation]

The concept of spreading activation has been quite popular in cognitive psychology (Collins & Loftus, 1975; Collins & Quillian, 1972; Kieras, 1977). There is an activation process that operates on the ACT network. Particular nodes can be activated either by stimulation from environmental events or by execution of production actions. With the passage of time activation spreads from the source nodes to associated structures. Productions, in matching their conditions, can only inspect the active portion of memory. Therefore, this activation process serves to focus attention. There are also mechanisms in ACT which deactivate all the structure that has been activated by a source when that source loses its activation.

A spreading activation process was suggested by Quillian [[27]]. There is considerable physiological evidence for an associative spread of excitation through the nervous system [[29]]. There is also considerable behavioral data pointing to the utility of the concept [[3, 11]]. For instance, subjects can be slowed in deciding a proposition is false if there is an irrelevant connection among the elements of the proposition (e.g., Madrid is in Mexico, see [[11]]). In other circumstances such irrelevant connections can facilitate processing [[23]]. Such effects indicate a diffuse activation process that is priming all connections, relevant or not.

It seems that Quillian's spreading activation notion has not received much acceptance in the AI community (see [[24, 34]] for criticisms). It is computationally expensive to compute a spread of activation. In fact, in our own simulation we have found it too expensive to simulate faithfully the conceptual-neural model we would want to endorse. Rather we approximate this with a much quicker activation process. We suspect that it is the cost of the activation computation that accounts for the infrequency of its use in AI. By application of (4) we might be tempted to conclude that the conceptual-neural model is bad cognitive psychology. However, this problem with spreading activation may reflect only current computational limits. Given faster processors and the prospect of parallel computation, we suspect that efficiency objections to the conceptual-neural model will be less serious with passing years.

The computational function of the activation process within ACT is to enable us to run a pattern-evoked production system in a propositional network but to avoid linear (or worse) growths in the amount of computation time with the size of the data base or the number of productions. This is achieved in two ways. First, the activation process serves to limit the amount of structure needing to be searched in determining whether the condition of a nominated production matches.

Second, productions are nominated for consideration by the activation process. Associated with each node is a list of productions that make reference to that node. When that node is activated these productions are considered. Since the number of nodes that are active is independent of the total number of productions, under this scheme the growth of computation time with number of productions will depend on the average number of productions that are associated with any given node. Newell (personal communication --see also [[22]]) has argued that in realistic problem domains the number of productions tends to increase more rapidly than the number of data base elements. If this were true of ACT, there would be an increase in the production to node ratio with growth in the size of the system. Newell has observed from simulations that this increase in the ratio is approximately logarithmic with the number of productions. We would regard a logarithmic complexity function as acceptable.

The activation process basically serves to limit the amount of data that the system has available at any one time. Therefore, it is

important to the operation of the system that the currently available set of data not be just randomly selected but rather be associated to those source nodes the system is currently focused on. These are the currently relevant data. For instance, when a word is heard it is made a source node, activation spreading from that node will activate syntactic and semantic information needed for its comprehension.

While the utility of activating the associative surround of items in focus is clear, the utility of stretching out this activation process over time may not be so obvious. Why not immediately activate all the structure that is going to be activated? The spreading process allows ACT to focus first on those data most likely to be relevant, and to try less relevant data later. Thus, the spreading process allows the system to "bet" its limited computational resources on what is most promising first.

This is facilitated by having strengths associated with network links. The strength of a link reflects how frequently and recently it has been involved in successfully matching the condition of a production. In the spread of activation, the total amount of activation energy at a node is divided among the links emanating from that node. The amount of energy given to a link is a function of its strength relative to the strength of all links. The rate at which activation will spread down a link to activate structures connected to the link is a function of this amount of energy. This means the structure that tends to be more rapidly activated is the structure that

has more often proven useful in matching productions. This, of course, is a computationally sensible criterion for ordering the activation of structure. It is also a well documented psychological fact [[3, 4]] that subjects can retrieve more quickly information which they have used often.

This strength mechanism is also responsible for forgetting in ACT. Forgetting of information occurs when the links encoding a structure become so weak relative to competing links that it becomes effectively impossible to reactivate the structure. This implies that forgetting is due to competition by interfering information--one of the most well-documented facts about human memory [[4]].

[Virtues of Production Systems]

Production systems by now are a familiar AI formalism and there are discussions available of the advantages that they offer (cf., [[14]]). Therefore we will focus mainly on the advantages production systems provide as a psychological theory and on those aspects of the ACT production system that are unique. A striking similarity has been noted [[3, 26]] between production systems and other stimulus-response(S-R) theories in psychology. The connection between condition and action is quite similar to the S-R connection. There have been a series of psychological critiques of S-R theory [[3, 4, 8, 10, 18]]. These critiques have basically been aimed at the computational power of the S-R formalism. It has been shown [[18]] that at least certain versions of S-R models do not have the computational

power of finite state machines. However, production systems circumvent these limitations by use of variables, patterns, and memory [[3]]. Thus, production systems can be seen as having some of the psychological advantages of S-R theory without forfeiting computational power. The positive features of production systems discussed below are also true of S-R theories.

[Data- Driven Character]

One of the important features of production systems is their data-driven character. In each cycle of the production system there must be a reevaluation of the consequences of the current knowledge state for the control of behavior. Thus information that comes in and changes the knowledge state can have immediate effects on behavior. The survival advantages of a data-driven processing system for humans are obvious. There is also evidence that human cognition, even when survival is not at stake, operates in a strongly data-driven mode. For instance, the work on chess [[9, 26]] indicates chess masters do much of their intellectual work by means of pattern recognition, and that chess masters usually perceive the correct move within a few seconds of scanning the board. It has been argued that chess masters' knowledge of board positions can be properly modeled by a large production system where individual productions contain board configurations as condition-patterns and appropriate responses (often moves) as actions.

[Unity of Control and Data Store]

One of the central features of production systems is their lack

of distinction between the medium that stores control information and the medium that stores data. Unlike most programming languages there are no special facilities for storing control information--no separate program counter, pushdown stack, etc. All control information must be stored in the same data base (e.g., in Newell's PS this is STM, in ACT it is the propositional network) that serves to store the input and results of computations.

Intuitively, it seems compelling to us that unity of control and data is a feature of human cognition. Fortunately, we do not have to rely just on intuition. There is some psychological data supporting this assumption. A variety of experiments have looked for a trade-off between the size of the immediate memory span and the computational complexity of an on-going task [[7, 13, 30, 33]]. For instance, Wanner and Maratsos [[33]] had subjects hold a set of words in memory while the subjects tried to comprehend sentences of various syntactic complexity. In the ATN model they were testing, this complexity translated into amount of control information that had to be held. (It would translate into amount of control information in most computational models.) They found subjects were able to recall fewer of the words when comprehending sentences of greater syntactic complexity, and hence requiring more control information. This trade-off between the amount of information in memory span and the amount of control information is typical of the research results in this area. The hypothesis that there is one storage medium for data and control information predicts this combined storage limitation.

Our feeling is that the use of a single representational formalism for control information and all other data has advantages for a system that is to acquire and modify productions. The role of an individual production in the overall flow of control becomes more apparent, and this comprehensibility should be as much of an advantage for a learning system which must make decisions about the function of a particular production as it is for a human programmer. The structure of productions and their rules of interpretation are also considerably simplified if there is no distinction between control information and other data, and this simplicity is also an advantage for production acquisition.

Unfortunately, while production systems tend not to make a formal distinction between control and data, in practice, one often finds an implicit distinction made by the programs written within production systems. This is seen in the use of special control conventions in order to achieve the same sequential execution of functions found in conventional programming languages. Although some of our earlier work in ACT had made extensive use of such control conventions, we are currently trying to avoid them because they forfeit some distinct advantages of production systems.

[Modularity]

Another important feature of production systems is the modularity of the individual productions. Because each production makes reference to a data base common to all productions and because no

production makes reference directly to other productions, individual productions tend to be independent of one another. That is, if a particular production is added, deleted, or changed, the basic performance of the system tends to remain relatively unaffected. We feel that claims about the modularity of production systems have been sometimes exagerated. There can be production systems designed such that the change of one production would have disastrous effects on the system. However, in a sense such production systems reflect "bad programming style." Production systems should be constructed to maximize the modularity of individual productions.

The advantage of modularity to the comprehensibility of the system is obvious. The advantage of modularity to the development of a learning system capable of self-extension is equally obvious. The basic modularity of human information processing is attested to by the gradual development of our processing abilities and by the fact that new skills almost never have disastrous interactions with old. (Of course, these facts about human modularity could be produced by formalisms other than production systems.)

[Parallelism]

There are a number of places in ACT where we have found it useful to simulate parallel computation. As discussed above, activation is conceived of as spreading simultaneously from all active nodes to the associated network structure and in the process, selecting the productions that reference these nodes for further testing. Not

only does ACT select in parallel among a large set of productions on each cycle, it is also the case that on each cycle ACT simulates the ability to apply (test and execute) a number of productions in parallel. The evidence is quite compelling that humans can carry along a number of processes in parallel. For instance, consider the well-worn example of being able to hold a conversation and drive at the same time. We have found parallel procedures useful in language processing [[5]]. In comprehending a sentence one must perform a large number of operations--make perceptual judgments about words, switch attention from word to word, perform syntactic analysis, perform semantic analysis, resolve pronominal references, recognize the referents of definite descriptions, perform inferences, etc. It is implausible that each of these operations occupies separate segments of time with control switching among them. It is much more intuitively plausible to suppose that these various linguistic processes proceed in parallel.

While we do allow ACT to pursue more than one process in parallel, there is a capacity limitation on how much can be computed in parallel. There is a probabilistic parameter that places a maximum on the mean number of productions applied per cycle. In the current implementation this parameter is (arbitrarily) set to limit this mean to 10 productions per cycle. As the number of productions required per cycle approaches this limit, performance degrades. This degradation is produced by making productions wait multiple cycles before applying and forcing certain processess (sequences of productions) to drop out. The

evidence is quite clear for such capacity limitations in the human case ( [[3]], Section 6.3). A familiar example is the problem of holding a conversation in difficult traffic. As the amount of computation required for processing traffic information increases, less capacity is left over for the conversation. The AI motivation of such a limitation is simply that the limits of finite processing capacity must be respected in any physical machine (whether it has parallel processors or not).

Given that only a small number of productions will be allowed to apply in parallel, it becomes important to have some way of deciding which of those that can apply are most relevant. This is very much like the problem discussed above of insuring that the currently most relevant nodes in the network are those that are allowed to be active. A solution in terms of strength measures is also adopted in this case. Associated with each production is a strength reflecting the past success of that production. Details of how this strength is computed are motivated by learning considerations and so have been omitted from this discussion. For present purposes it is important only to note that the probability that a production will apply on a cycle when its condition is satisfied depends on the strength of that production relative to the strengths of all the other productions whose conditions are also satisfied. This means that the stronger productions (and hence the more successful processes) tend to be tried first and are less disturbed by processes going on concurrently.

Consider the implication of these strength mechanisms for processing of structural ambiguity in language where competing sets of productions are responsible for different interpretations of the same linguistic structure. The relative frequency of the two structural interpretations will be reflected by the relative strengths of the two sets of interpretative productions. We would therefore expect that subjects would tend to interpret the sentence according to the more frequently intended interpretation, but that they could retrieve the other interpretation and that they would sometimes choose the less frequent interpretation first. Also, we would expect that by bombarding the subject with examples of the less frequent interpretation we could make it the favored interpretation of the ambiguous structure. All these implications are known to be true [[15]].

## Footnotes

1. Distribution of this manuscript and current research on the ACT model are supported by ONR contract N00014-77-C-0242.

2. It has to be pointed out here that there is still considerable ignorance about the capabilities of serial computers and their abstractions. For instance, while many problems are conjectured to have exponent time functions as minimums, no natural problem has been shown to require greater than $n^{1+E}$ for a E>0 [[1]].

[References]


1. Aho, A.V., Hopcroft, J.E., & Ullman, J.D. [The Design and Analysis of Computer Algorithms.] Reading, Mass.: Addison-Wesley, 1974.

2. Anderson, J.R. Language acquisition by computer and child. Technical Report No. 55, Human Performance Center, 1974.

3. Anderson, J.R. [Language, Memory, and Thought]. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976.

4. Anderson, J.R. and Bower, G.H. [Human Associative Memory.] Washington, D.C.: V.H. Winston, 1973.

5. Anderson, J.R., Kline, P. and Lewis, C. A production system model for language processing. In P. Carpenter and M. Just (Eds.) [Cognitive Processes in Comprehension.] Hillsdale, N.J.: Lawrence Erlbaum Assoc., in press

6. Anderson, R.C. Substance recall of sentences. [Quarterly Journal of Experimental Psychology], [26], 1974, 530-541.

7. Baddeley, A.D. and Hitch, G. Working memory. In G.H. Bower (Ed.) [The Psychology of Learning and Motivation], [8], 1974, New York: Academic Press.

8. Bever, T.G., Fodor, J.A., and Garrett, M. A formal limitation of associationism. In T.R. Dixon and D.L. Horton (Eds.) [Verbal Behavior and General Behavior Theory]. Englewood Cliffs, N.J.: Prentice-Hall, 1968.

9. Chase, W.G. and Simon, H.A. The mind's eye in chess. In W.G. Chase (Ed.), [Visual Information Processing]. New York: Academic Press, 1973.

10. Chomsky, N. Verbal behavior (a review of Skinner's book). [Language], [35], 1959, 26-58.

11. Collins, A.M. & Loftus, E.F. A spreading- activation theory of semantic processing. [Psychological Review], [82], 1975, 407-428.

12. Collins, A.M. & Quillian, M.R. Experiments on semantic memory and language comprehension. In L. Gregg (Ed.) [Cognition and Learning]. New York: Wiley, 1972.

13. Crowder, R.G. Behavioral strategies in immediate memory. [Journal of Verbal Learning and Verbal Behavior], [8], 1969, 524-528.

14. Davis, R. and King, J. An overview of production systems. Computer Science Department, Stanford University, 1975.

15. Fodor, J.A., Bever, T.G., and Garrett, M.F. [The Psychology of Language]. New York: McGraw-Hill, 1974.

16. Fredericksen, C. Effects of context-induced processing operations on semantic information acquired from discourse. [Cognitive Psychology], [7], 1975, 139-166.

17. Johnson, D.M. [A Systematic Introduction to the Psychology of Thinking]. New York: Harper and Row, 1972.

18. Kieras, D.E. Finite automata and S-R models. [Journal of Mathematical Psychology], [3], 1976, in press.

19. Kieras, D.E. Problems of reference in text comprehension. In P. Carpenter and M. Just (Eds.) [Cognitive Processes in Comprehension]. Hillsdale, N.J.: Lawrence Erlbaum Assoc., in press.

20. Kiss, G.R. A test of the word selection model using multiple stimuli in word association. Paper presented at the conference of the British Psychological Society, London, 1967.

21. Kline, P. and Anderson, J.R. The ACTE User's Manual. Dept. of Psychology, Yale University, 1976.

22. McDermott, J., Newell, A., and Moore, J. The efficiency of certain production system implementations. Dept. of Computer Science, Carnegie Mellon University, Sept., 1976.

23. Meyer, D.E. & Schvaneveldt, R.W. Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. [Journal of Experimental Psychology], [90], 1971, 227-234.

24. Minsky, M. [Semantic Information Processing]. Cambridge, Mass.: The M.I.T. Press, 1968.

25. Newell, A. Remarks on the relationship between artificial intelligence and cognitive psychology. In R.B. Banerji and M.D. Mesarovic (Eds.), [Theoretical Approaches to Non-Numerical Problem Solving]. Berlin: Springer-Verlag, 1970.

26. Newell, A. and Simon, H. [Human Problem Solving]. Englewood Cliffs, N.J.: Prentice-Hall, 1972.

27. Quillian, M.R. Semantic memory. In M. Minsky (Ed.) [Semantic Information Processing], Cambridge, Mass.: MIT Press, 1968.

28. Sachs, J. Recognition memory for syntactic and semantic aspects of connected discourse. [Perception and Psychophysics], [2], 1967, 437-442.

29. Thompson, R.F. [Foundations of Physiological Psychology], New York: Harper & Row, 1967.

30. Savin, H.B. and Perchonock, E. Grammatical structure and the immediate recall of English sentences. [Journal of Verbal Learning and Verbal Behavior], [4], 1965, 348-353.

31. Simon, H.A. [The Sciences of the Artificial]. Cambridge, Mass.: MIT Press, 1969.

32. Wanner, H.E. On remembering, forgetting, and understanding sentences: A study of the deep structure hypothesis. Unpublished doctoral dissertation, Harvard University, 1968.

33. Wanner, E. and Maratsos, M. An augmented transition network model of relative clause comprehension, unpublished manuscript, Harvard University, 1975.

34. Winograd, T. Procedures as a representation for data in a computer program for understanding natural language. [MIT Artificial Intelligence Laboratory Project, MAC-TR-84], 1971.

35. Woods, W.A. What's in a link: Foundations for semantic networks. In D.G. Bobrow and A. Collins (Eds.), [Representation and Understanding: Studies in Cognitive Science]. New York: Academic Press, 1975.